Actividades Cliente-Servidor TCP monohilo

Actividad 1

Esta actividad consiste en **analizar el código fuente que se te proporciona**. En concreto los ficheros *EcoCliente.java* y *EcoServidor.java*.

Se trata de un sencillo ejemplo de aplicación cliente-servidor de eco, es decir, el servidor repite lo mismo que le envía el cliente, hasta que el cliente quiere finalizar el servicio, para lo cual envía la palabra "Adios" al servidor.

Centrate en la parte de comunicaciones y observa la forma general de implementar un cliente y un servidor.

Forma de implementar un **cliente**:

1. Crear un objeto de la clase Socket, indicando host y puerto donde se está ejecutando el servicio.
2. Obtener las referencias al stream de entrada y de salida del socket
3. Leer desde y escribir en el stream de acuerdo al protocolo de servicio. Para ello emplear algunas de las facilidades del paquete java.io.
4. Cerrar los streams
5. Cerrar el socket.

Forma de implementar un **servidor**:

1. Crear un objeto de la clase ServerSocket para escuchar peticiones en el puerto asignado al servicio.
2. Esperar solicitudes de clientes
3. Cuando se produce una solicitud:
   - Aceptar la conexión obtreniendo un objeto de la clase Socket
   - Obtener las referencias al stream de entrada y al de salida al socket anterior
   - Leer datos del socket, procesarlos y enviar respuestas al cliente, escribiendo en el stream del socket. Para ello emplear alguna de las facilidades del paquete java.io.
4. Cerrar los streams
5. Cerrar el socket.

Observa que el interfaz que da soporte a sokets TCP está consitutido por las clases ServerSocket y Socket

1. **ServerSocket**: es utilizada por un servidor para crear un socket en el puerto en el que escucha las peticiones de conexión de los clientes. Su método accept toma una petición de conexión de la cola, o si la cola está vacía, se bloquea hasta que llega una petición. El resultado de ejecutar accept es una instancia de Socket, a través del cual el servidor tiene acceso a los datos enviados por el cliente.
2. **Socket**: es utilizada tanto por el cliente como por el servidor. El cliente crea un socket especificando el nombre del host y el puerto del servidor, así se crea el socket local y además se conecta con el servicio. Esta clase proporciona los métodos getInputStream y getOuptutStream para acceder a los dos streams asociados a un socket (recuerda que son bidireccionales) y devuelve tipos de datos InputStream y OutputStream, respectivamente, a partir de los cuales podemos construir BufferedReader y PrintWriter, respectivamente, para poder procesar los datos de forma más sencilla.

Para probar como funciona este sencillo programa, debereis arrancar primero el servidor, que se quedará a la escucha de peticiones, y después el cliente. Si lo haceis en orden inverso, al arrancar el cliente se producirá una IOException.

Modifica el programa anterior para que el servidor pueda atender peticiones de varios clientes.